Do Motion Boundaries Improve Semantic Segmentation?

Yu-Hui Huang[◇], José Oramas M.[◇], Tinne Tuytelaars[◇] and Luc Van Gool[◇∓]
[◇]ESAT-PSI, KU Leuven [∓]CVL, ETH Zürich.
{yu-hui.huang,jose.oramas,tinne.tuytelaars,luc.vangool}@esat.kuleuven.be

Abstract

Precise localization is crucial to many computer vision tasks. Optical flow can help by providing motion boundaries which can serve as proxy for object boundaries. This paper investigates how useful these motion boundaries are in improving semantic segmentation. As there is no dataset readily available for this task, we compute the motion boundary maps with a pre-trained model from [17] on the CamVid dataset [3]. With these motion boundary maps and the corresponding RGB images, we train a convolutional neural network end-to-end, for the task of semantic segmentation. The experimental results show that the network has learned to incorporate the motion boundaries and that these improve the object localization.

1 Introduction

Object localization is of crucial importance for several applications, e.g. autonomous navigation where the precise delineation of objects can serve as input for obstacle avoidance and lane detection. Over the years this problem has been addressed from several perspectives, e.g. semantic segmentation [18], object detection [10], and scene 3D reconstruction [15]. Despite the efforts, the accurate delineation of object boundaries remains a challenge, given that object outlines are confounded by various appearance-related factors, like self/inter-object occlusions or shadows.

In this paper we propose to leverage motion cues, in the form of motion boundaries [17], to alleviate the effect of the previous factors. Motion boundaries are derived from the abrupt changes in optical flow. They reveal the location of occlusions and object boundaries. Motion boundaries have been widely used in different computer vision tasks such as action recognition [16], and object delineation in videos [12]. In this paper, we investigate the potential of motion boundaries at improving semantic segmentation. We propose to concatenate the motion boundary map to the original RGB image, and use this to train a network for semantic segmentation. Our main hypothesis is that the inclusion of motion boundaries can improve semantic segmentation performance. Regarding the integration of motion cues for semantic segmentation, Sevilla-Lara et al. [13] used localized layers to iteratively improve both semantic segmentation and optical flow. Tokmakov et al. [14] trained a network using motion segments as a soft constraint for semantic segmentation. These methods already bring significant improvements compared to methods that do not employ motion cues, but their optimization steps are time consuming. To alleviate this issue, we choose a simple yet efficient convolutional neural network (CNN) architecture, SegNet, proposed by Badrinarayanan et al. [1].

The paper is organized as follows. Section 2 presents related work. In Section 3 we present the proposed method. We introduce the motion boundary detector used in our experiments and describe our extended version of SegNet. In Section 4 we present our evaluation protocol followed by experimental results and discussions. Finally, Section 5 concludes this paper.

2 Related Work

Semantic segmentation. Long et al. [11] were the first to propose an end-to-end convolutional neural network for semantic segmentation. They modified the last layers of the CNN, thus producing a fully convolutional neural network (FCN). Due to the large receptive fields of FCNs, the localization of object boundaries is insufficiently precise. Different remedies were proposed, such as applying fully connected CRFs to the output of the CNN [6] or introducing a global energy model integrating boundary cues [2] to improve the segmentation accuracy near object boundaries. Such post-processing steps require additional parameter tuning, however. Recently, Badrinarayanan et al. [1] proposed an encoder-decoder based architecture called SegNet. Compared to FCNs, SegNet requires one fourth of memory usage and about half of inference time, making it an ideal architecture for efficient segmentation.

Motion cues. Papazoglou et al. [12] derived motion boundaries from the gradient of optical flow and the angle of the flow. Weinzaepfel [17] proposed a learning based prediction method using a structured random forest [7]. They combine temporal and spatial cues from optical flow and local features to learn a detector. Given the quality of their results, we decided to adopt their method to generate motion boundaries, as additional input for our semantic segmentation model.

3 Proposed Method

Our method contains three steps. Given an image sequence, it first computes the optical flow for the whole sequence. Secondly, it computes the motion boundary maps from the optical flow. Thirdly, the motion boundary maps are concatenated with their corresponding RGB images, as input to a deep convolutional neural network to predict frame-level segmentation maps. We present these main components in the following sections.

3.1 Motion Boundary

Given an image sequence, we use an off-the-shelf pre-trained detector from [17] to predict motion boundaries. This detector was trained on the MPI-Sintel dataset [5] using a structured random forest [7]. It exploits the characteristic that motion boundaries look similar in local patches, and uses this to predict the input patch as a structured output. Motion boundaries are predicted by using a combination of static appearance features and temporal features. The static appearance features come from the three RGB channels and ten gradient maps derived from the luminance channel of the Lab color space. Temporal features are composed of forward/backward optical flow and corresponding image warping errors. The forward optical flow is computed from the current to the next frame while the backward flow is computed from the current to the previous frame. In addition to u,v (displacement) channels, they include both unoriented and oriented gradient maps. Image warping errors are measured based on the level to which the gradient and color (in Lab color space) constancy assumption [4] is violated, using Euclidean distance.

For a given input image, we uniformly sample image patches using a sliding window. Each patch is then subsampled, and for each of them we compute the features mentioned above. After that, the concatenated features from each patch are fed into the structured random forest to predict the corresponding binary mask. The resulting binary masks are aligned to the original patch using the edge sharpening technique from [7] and then averaged to yield the final soft-response boundary map.

3.2 Semantic Segmentation

Instead of refining the segmentation results using a post-processing step, we aim to improve the delineation by utilizing the motion boundaries. Our method is based on the SegNet [1] architecture, which offers a good tradeoff between segmentation performance and efficiency in terms of memory and computation time. SegNet is an encoder-decoder architecture composed of sequences of encoders and corresponding decoders. Each encoder contains convolution, batch normalization and element-wise rectified-linear nonlinearity (ReLU) components. At the end of each encoder follows a max pooling layer to achieve more translation invariance and the max-pooling indices are stored for the subsequent stage. The decoder starts with upsampling using the stored max-pooling indices and then performs sequences of convolution, batch normalization and ReLU components.

To integrate motion boundaries into the network, we propose to concatenate the motion boundary maps calculated in the previous section to their corresponding RGB images to train a SegNet-like model. Instead of using the motion boundaries as an additional modality in a late fusion post-processing step [2], we propose to integrate motion boundaries as part of the input, thus allowing the network to learn from them.

During testing, given an image sequence, we first compute the motion boundary maps using consecutive frames. Then we concatenate the boundary map with its corresponding RGB image, and feed them to our network which produces a segmentation map as output.

4 Experiments

Dataset. We have conducted experiments on the CamVid dataset [3]. Although it is not a big dataset, it contains some quite challenging scenarios such as road scenes in the dusk and objects at a small scale. There are in total 12 classes including the *empty* class. For training we use the defined training set (367 images) and we test on the test set (233 images) at a resolution of 480x360 pixels.

Implementation Details. From the input sequences we first compute the backward/forward optical flow using the pre-trained FlownetC network [9]. With the optical flow maps and corresponding RGB images for three consecutive frames, we compute the motion boundary map for the middle frame with the pre-trained detector from [17]. We sample image patches of 32 x 32 pixels with a stride of 2 pixels and then subsample them by a factor of 2. For training a SegNet, we use stochastic gradient descent (SGD) with a fixed learning rate of 0.1 and momentum of 0.9 until the training loss converges. For training we use the cross-entropy loss. At each training epoch, the training set is shuffled and images are picked using a mini-batch of 6 images. As the number of samples in each class is not balanced, we follow [1] to use median frequency balancing [8] as a weight to calculate the loss value.

We compare our method to the SegNet model from [1]. To this end, we train a SegNet from scratch, following the parameters provided in [1], on the CamVid training set. This model will constitute the baseline (denoted as SegNet in Table 1) of our first experiment and will show the performance that can be obtained when using RGB data only. In addition, we train our proposed method (+Motion-Boundary) where we extend the input by concatenating the motion boundary map described previously with the RGB

Method	Global avg.	Class avg.	Mean I/U
SegNet	78.5	54.8	41.3
+3edges	75.5	**57.9**	40.9
+OpticalFlow	76.4	56.9	41.2
+MotionBoundary	**79.1**	56.5	**43.1**

Table 1: Quantitative results on the CamVid.

image. For the sake of a fair comparison, we define an extension (+3edges) of the baseline method, by concatenating the edge maps [7] from the three neighboring frames to the original input. This way, extended SegNet method, +3edges, considers the same frame information as the proposed method. Finally, in order to see whether other types of motion representation are effective, we train a SegNet concatenating the RGB data with the optical flow map calculated by FlowNetC instead of with the motion boundary map. The results can be found in Table 1.

In the second experiment, we initialize the weights of the SegNet encoder, related to the RGB data, using the VGG16 model pre-trained on ImageNet. This is done for our baseline and our proposed method. Since the input of our proposed model is different from the pre-trained VGG model, the filter size of the first convolutional layer is not identical. Thus, we copy the filter related to the motion channel (including its weights) from the model trained in the previous experiment. This way we compensate for the difference w.r.t. the pre-trained VGG model. See Table 2 for quantitative results.

We report quantitative results using global average, class average and mean intersection over union (mean I/U) as performance metrics. Global average is the percentage of pixels correctly classified over the entire images, class average is the mean of accuracy from all classes and mean I/U is determined by the number of true positives divided by both positives and false negatives.

4.1 Discussion

A quick inspection of Table 1 shows that integrating motion boundaries can improve semantic segmentation. This is supported by improvements over the original SegNet architecture [1] It seems that by utilizing rich edge information (+3edges) can improve class average performance at the cost of false positives which is reflected in the drop of mean I/U performance. In this last metric, using motion boundaries still brings some levels of improvement. Moreover, using motion boundaries seems to be an effective means to encode motion information, producing improvements over optical flow, especially on the mean I/U metric which gives stronger emphasis to false positive predictions. From the segmentation results in Figure 1, our method seems to also perform better qualitatively than the baseline. Visual inspection suggests that the object boundaries are more precise. For instance, the car in the top row has been delineated better, because of the information added by the motion boundary maps. In addition, we can notice that small-scale objects, such as *bicyclists*, are better defined.

When considering the SegNet with pre-trained encoder (initialized on VGG 16), we can notice that our method is 1.2% inferior in global average, 4 % superior in class average and 0.7 % superior in mean I/U. This can be attributed to several factors. First, for the case of global average, this metric is dominated by classes with large spatial extent, e.g. *sky, road, building*, which are present in our dataset. The baseline method achieves higher performance in this metric which suggests that it is good at segmenting large extent classes. In contrast, the class average and mean I/U metrics reduce the dominance of classes with a large spatial extent by normalizing with respect to the number of pixels in each class. In this regard, the baseline method has lower performance. Given that object boundaries are the places where pixel class changes, and that these metrics are affected by the pixel class, we can infer that the proposed method is superior to the baseline at this boundary case. This suggests that considering motion boundaries can help improving segmentation at the location of object boundaries.

From Table 2, the proposed method shows a superior performance on classes with small instances, e.g. *fence, pole, bicyclist, tree, etc.* with exception of the class *sign*. This exception can be explained by the fact that far away signs appear at very low scale due to projection effect. Optical flow fails at this scenario which is confirmed by its performance. Since our proposed method depends on the quality of optical flow, it inherits this weakness, yet still manages to achieve superior performance than +OpticalFlow on this hard scenario. This suggests that integrating motion boundaries has some potential in improving segmentation of small-scale classes.

As the experimental results show, training a SegNet with motion boundaries improves the performance of semantic segmentation. Even though our method outperforms the baseline in some aspects, there is room for improvement. First, we propose to use a more precise optical flow algorithm, perhaps by considering more consecutive frames or higher resolution images. Second, another direction is to integrate the motion cues within the end-to-end learning framework.

5 Conclusion

We presented a simple way to include motion boundaries into semantic segmentation. Our experimental results suggest that the inclusion of motion boundary maps can improve semantic segmentation. Furthermore, our results show that motion boundaries have the potential to improve segmentation in two cases of object boundary locations and smaller-scale objects.

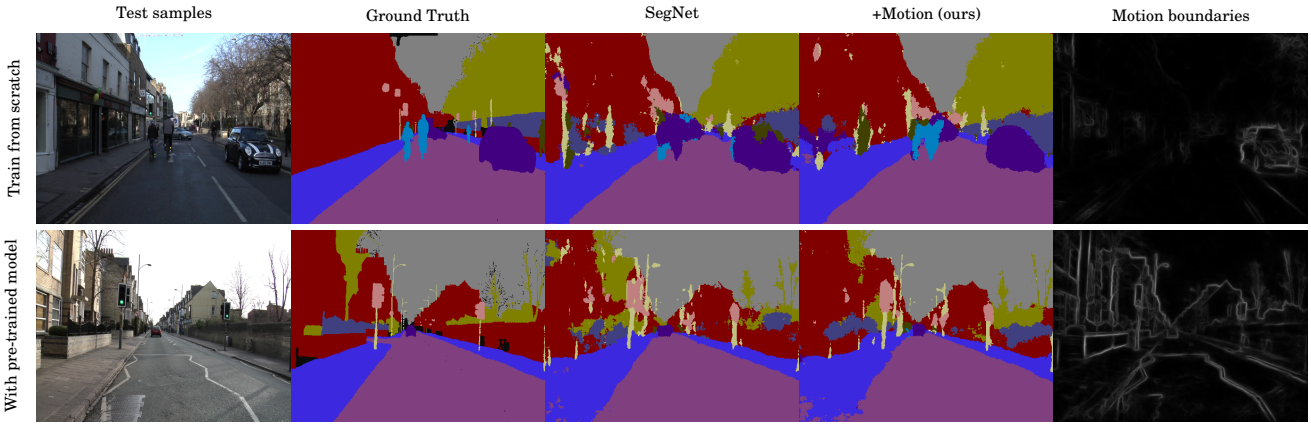

Figure 1: Qualitative results from CamVid dataset. Best visualized in digital form.

	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	Mean I/U	Global avg.	Class avg.
SegNet	75.6	68.0	88.8	71.5	**32.4**	**90.7**	36.0	25.6	20.9	73.7	27.9	55.6	**88.5**	65.2
+OpticalFlow	74.3	67.5	89.3	**72.6**	26.8	90.6	36.2	28.7	**22.8**	**73.8**	29.0	55.6	86.7	**70.6**
+MotionBoundary	**76.0**	**69.6**	**89.5**	70.3	31.3	90.3	**36.6**	**29.6**	22.7	72.8	**30.6**	**56.3**	87.3	69.4

Table 2: Performance in per-class I/U when initializing the evaluated method with the pre-trained VGG model on CamVid.

Acknowledgments: This work is supported by the European project EUROPA2, a bilateral grant with Toyota, and NVIDIA Academic Hardware Grant.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv:1511.00561*, 2015.

[2] G. Bertasius, J. Shi, and L. Torresani. Semantic Segmentation with Boundary Neural Fields. In *CVPR*, 2016.

[3] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic Object Classes in Video: A High-Definition Ground Truth Database. *Pattern Recognition Letters*, 2008.

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *ECCV*, 2004.

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A Naturalistic Open Source Movie for Optical Flow Evaluation. In *ECCV*, 2012.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *ICLR*, 2016.

[7] P. Dollár and C. L. Zitnick. Structured Forests for Fast Edge Detection. In *ICCV*, 2013.

[8] D. Eigen and R. Fergus. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. In *ICCV*, 2015.

[9] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *ICCV*, 2015.

[10] J. Hosang, M. Omran, R. Benenson, and B. Schiele. Taking a Deeper Look at Pedestrians. In *CVPR*, 2015.

[11] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*, 2015.

[12] A. Papazoglou and V. Ferrari. Fast Object Segmentation in Unconstrained Video. In *ICCV*, 2013.

[13] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical Flow with Semantic Segmentation and Localized Layers. In *CVPR*, 2016.

[14] P. Tokmakov, K. Alahari, and C. Schmid. Weakly-Supervised Semantic Segmentation using Motion Cues. In *ECCV*, 2016.

[15] V. Usenko, J. Engel, J. Stückler, and D. Cremers. Reconstructing Street-Scenes in Real-Time From a Driving Car. In *3DV*, 2015.

[16] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *IJCV*, 103(1):60–79, 2013.

[17] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Learning to Detect Motion Boundaries. In *CVPR*, 2015.

[18] C. Zhang, L. Wang, and R. Yang. Semantic Segmentation of Urban Scenes Using Dense Depth Maps. In *ECCV*, 2010.